AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2) (2026)

Design and Implementation of an AI exam invigilator

Mr. Jatin Kumar Bhuyan
Student, Dept. of CSE(AI),
GIFT Autonomous, Bhubaneswar

Mr. Diptesh kumar Sahoo
Student, Dept. of CSE(AI),
GIFT Autonomous, Bhubaneswar

Mr. Swarnananda Muduli
Assistant Professor, Dept. of CSE(AI),
GIFT Autonomous, Bhubaneswar

***Abstract*— The rapid growth of online and computer-based examinations has increased the demand for reliable, scalable, and intelligent examination monitoring systems. Traditional invigilation methods rely heavily on human supervisors, which can be inefficient, costly, and prone to human error. This project presents the design and implementation of an Artificial Intelligence (AI)-based Exam Invigilator system that automates the process of monitoring candidates during examinations. The proposed system utilizes advanced AI techniques such as computer vision, facial recognition, eye movement tracking, object detection, and audio analysis to detect suspicious activities and ensure examination integrity. A webcam continuously captures video streams of examinees, while machine learning algorithms analyze facial orientation, multiple person presence, unusual head movements, mobile phone usage, and absence from the screen. The system can generate real-time alerts and maintain logs of detected malpractice incidents for later review by administrators.**

Keywords*— *Python, OpenCV, deep learning frameworks, and cloud-based storage to provide an efficient and scalable solution

I. INTRODUCTION

The advancement of digital technology and internet connectivity has significantly transformed the education sector, leading to the widespread adoption of online learning and computer-based examinations. Educational institutions, universities, and certification organizations increasingly rely on remote examination systems to provide flexible and accessible assessment methods. However, maintaining the integrity, fairness, and credibility of online examinations remains a major challenge due to the increased risk of cheating and malpractice. Traditional invigilation methods, which depend on human supervisors, are often expensive, time-consuming, and less effective in large-scale or remote examination environments..

Artificial Intelligence (AI) has emerged as a powerful solution for automating monitoring and decision-making processes in various domains, including education. AI-based exam invigilation systems use technologies such as computer vision, machine learning, facial recognition, and behavior analysis to monitor candidates during examinations in real time. These systems can automatically identify suspicious activities such as unauthorized person presence, frequent head movements, eye diversion, mobile phone usage, speaking during exams, and absence from the examination screen. By providing continuous monitoring and automated alerts, AI invigilators help reduce human effort while improving examination security and reliability.

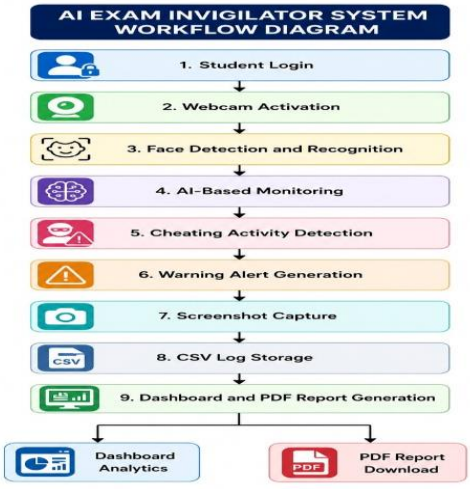
The "Design and Implementation of an AI Exam Invigilator" focuses on developing an intelligent system capable of monitoring students during online examinations and detecting possible malpractice activities. The proposed system uses a webcam and audio input devices to collect real-time data, which is processed using AI algorithms for behavior analysis and anomaly detection. The system records suspicious events, generates reports, and provides administrators with evidence for review.

The implementation of the AI exam invigilator involves integrating technologies such as Python programming, OpenCV for image processing, deep learning models for facial and object detection, and database systems for storing examination records and alerts. The system aims to provide a secure, scalable, and cost-effective examination monitoring solution suitable for schools, universities, and professional certification platforms.

This project contributes to the modernization of examination management by enhancing transparency, minimizing cheating opportunities, and supporting fair assessment practices in online education environments. The proposed AI invigilation system demonstrates how intelligent automation can improve the efficiency and effectiveness of remote examination supervision.

Figure-1:

The AI Exam Invigilator System follows a structured real-time monitoring workflow designed to ensure secure and intelligent online examination supervision. The system begins with secure student authentication and continuously analyzes webcam frames using Artificial Intelligence and Computer Vision technologies.



The AI Exam Invigilator System follows a modular layered architecture consisting of frontend, backend, AI processing

modules, and log management components.

II. EXISTING APPROACHES

Several approaches have been developed to monitor examinations and reduce malpractice in both traditional and online learning environments. These approaches range from manual supervision methods to advanced AI-powered proctoring systems. Each method provides certain advantages but also faces limitations in terms of efficiency, accuracy, scalability, and cost.

1. Traditional Human Invigilation

The most common examination monitoring method is human invigilation, where supervisors physically observe students during examinations. In classroom-based exams, invigilators monitor student behavior, verify identities, and prevent cheating activities such as copying or the use of unauthorized materials.

Although this method is simple and widely used, it has several limitations:

- Requires a large number of supervisors for large-scale examinations.
- Human errors and distractions can reduce monitoring effectiveness.
- Difficult to implement effectively in remote or online examinations.
- Increases operational costs and administrative workload.

2. Online Proctoring Systems

With the growth of online education, online proctoring systems were introduced to supervise remote examinations. These systems typically use webcams, microphones, and screen-sharing technologies to monitor students during exams. Some systems record video sessions for later review by human examiners.

Common features include:

- Candidate identity verification.
- Browser lockdown to prevent access to unauthorized websites.
- Screen activity monitoring.
- Audio and video recording during exams.

Despite improving remote supervision, many online proctoring systems still depend heavily on human reviewers, making them time-consuming and expensive.

3. AI-Based Proctoring Approaches

Recent developments in Artificial Intelligence and Machine Learning have enabled the creation of automated invigilation systems. AI-based approaches use computer vision and deep learning techniques to analyze candidate behavior in real time. These systems can detect suspicious activities such as:

- Multiple persons appearing on screen.
- Frequent head or eye movement.
- Mobile phone detection.
- Speaking or unusual sounds.
- Face absence from the camera view.

AI-based systems significantly reduce the need for continuous human monitoring and improve scalability. However, some existing solutions may produce false alerts due to lighting conditions, camera quality, or variations in user behavior.

4. Biometric Authentication Systems

Some examination systems use biometric technologies such as facial recognition, fingerprint scanning, or voice recognition to verify student identity before and during examinations. These systems help prevent impersonation and unauthorized access.

Limitations of biometric systems include:

- High implementation cost.
- Privacy and data security concerns.
- Dependence on high-quality hardware and stable network connectivity.

5. Hybrid Monitoring Systems

Hybrid systems combine human supervision with AI-powered monitoring tools. AI algorithms perform automated detection and flag suspicious incidents, while human invigilators review the alerts and make final decisions. This approach improves monitoring efficiency and reduces false accusations.

Although hybrid systems offer better accuracy, they still require trained personnel and can be expensive for institutions with limited resources.

LAYER	TECHNOLOGY	DESCRIPTION	PURPOSE
Frontend	HTML	Markup language	Structure of web pages
Frontend	CSS	Styling language	UI design and layout
Frontend	Bootstrap	CSS framework	Responsive design
Frontend	JavaScript	Scripting language	Dynamic functionality
Backend	Python	Programming language	Backend processing and AI integration
Backend	Flask	Web	Server-

LAYER	TECHNOLOGY	DESCRIPTION	PURPOSE
		framework	side routing and request handling
AI Processing	OpenCV	Computer vision library	Face detection and recognition
AI Processing	YOLOv8	Object detection model	Mobile phone detection
AI Processing	MediaPipe	AI face tracking framework	Head movement and face tracking
AI Processing	Haar Cascade	Face detection algorithm	Real-time face localization
AI Processing	LBPH Recognizer	Face recognition algorithm	Student face identification
Database/Storage	CSV	File-based storage system	Activity log management
Reporting	ReportLab	PDF generation library	PDF report creation

Table-I: Comparison of Traditional System and Proposed system

1.1.1. Frontend: HTML, CSS, JavaScript

The frontend of the AI Exam Invigilator System is developed using HTML, CSS, Bootstrap, and JavaScript to create a modern, responsive, interactive, and user-friendly monitoring interface.

The frontend provides smooth user interaction and real-time visual monitoring features for both students and administrators.

Frontend Features:

- Responsive login and dashboard interface
- Real-time webcam monitoring screen
- Interactive admin dashboard
- Dynamic warning alert display
- Glassmorphism-based UI design
- Responsive monitoring cards
- Modern futuristic dashboard layout
- Real-time cheating status visualization
- Screenshot preview interface
- Mobile-friendly responsive layouts
- Dynamic activity log display
- Interactive buttons and navigation system
- Real-time dashboard analytics visualization

Bootstrap ensures compatibility across desktops, laptops, and tablets, while JavaScript improves interactivity, dynamic updates, and real-time frontend responsiveness.

1.1.2. *Backend: Flask & Python*

The backend of the AI Exam Invigilator System is implemented using Flask and Python. The backend acts as the core processing engine responsible for AI integration, server-side operations, authentication, and report generation.

Backend Functions:

- Handling user authentication
- Managing student and admin login sessions
- Webcam stream handling
- AI module integration
- Real-time frame processing
- Face recognition processing
- Mobile detection execution
- Screenshot management
- CSV log generation
- Dashboard data processing
- PDF report generation
- Route and request handling
- Activity monitoring logic
- Session and security management

The backend ensures secure communication, smooth monitoring operations, real-time AI execution, and efficient system performance.

PROPOSED SYSTEM ARCHITECTURE

The AI Exam Invigilator System uses CSV-based storage and local file management for storing monitoring data, screenshots, trained models, and generated reports.

The storage layer of the AI Exam Invigilator System provides structured CSV log storage, automatic activity recording, screenshot evidence storage, PDF report storage, trained face recognition model storage, fast log retrieval, lightweight data management, secure evidence management, easy report generation support, and scalable storage architecture. The storage system maintains monitoring consistency and efficiently supports all major modules including logs, screenshots, reports, and AI models

.The system is developed using modern web technologies including React.js, Tailwind CSS, Flask REST APIs, SQL Alchemy ORM, PostgreSQL, and WebSocket-based real-time communication technologies. The frontend layer provides an interactive and responsive user interface for passengers, drivers, and administrators, while the backend layer handles business logic, ride allocation, fare calculation, real-time synchronization, authentication, and secure API

communication. The database layer securely stores user profiles, ride information, driver details, trip history, payment records, and operational analytics [20].

- head movement,
- looking away from the screen,
- mobile phone usage,
- multiple person presence,
- face absence,
- abnormal face positioning.

Whenever suspicious activity is detected, the system:

- displays warning alerts,
- captures screenshots automatically,
- stores logs in CSV format,
- updates dashboard analytics,
- The project integrates OpenCV, YOLOv8, MediaPipe, Flask, HTML, CSS, JavaScript, and Python to provide intelligent monitoring, responsive user interaction, and automated reporting capabilities.
- Unlike traditional online examination systems that mainly depend on manual supervision, the proposed system introduces automated AI-based monitoring and intelligent evidence generation, improving examination security and reducing malpractice.

Although many online examination and AI-based proctoring systems are available, several limitations still exist.
Common limitations include:

- limited cheating detection accuracy,
- dependency on manual monitoring,
- high system cost,
- poor real-time analysis,
- weak evidence generation,
- limited scalability,
- high hardware requirements,
- privacy concerns,
- lack of intelligent dashboard systems.

Many systems only record webcam video without performing intelligent analysis. Some systems also fail to generate proper logs, screenshots, or automated reports for administrators.
Additionally, advanced AI proctoring platforms often require expensive cloud infrastructure and high-performance hardware, making them difficult to adopt in smaller educational institutions.

Artificial Intelligence (AI) and Computer Vision technologies provide advanced solutions for automating online invigilation and improving examination security. AI-powered monitoring systems can continuously analyze student behavior in real time through webcam-based observation and detect suspicious activities automatically.

The AI Exam Invigilator System is developed to address these challenges by creating an intelligent AI-based online examination monitoring platform. The system uses technologies such as OpenCV, YOLOv8, MediaPipe, Flask, and Python to monitor students during online examinations and generate automated alerts, screenshots, logs, and reports whenever suspicious activities are detected.The modular architecture of the system provides several advantages including:

- Improved scalability and maintainability
- Secure API communication
- Centralized vulnerability monitoring
- Efficient vulnerability prioritization
- Interactive dashboard visualization
- Traditional online examination platforms often fail to provide intelligent behavior analysis and automated cheating detection mechanisms. Manual supervision becomes difficult and inefficient, especially during large-scale online examination

- **METHODOLOGY**

Online examination systems have become an important component of modern digital education platforms. These systems allow students to appear for examinations remotely using internet-based applications and virtual learning environments. Educational institutions increasingly use online examination platforms due to their flexibility, accessibility, and ability to support remote learning.

- question paper management,
- online test conduction,
- timer management,
- automatic result generation,
- student authentication.

The methodology of the AI Exam Invigilator System focuses on real-time intelligent monitoring using Artificial Intelligence and Computer Vision technologies. The system continuously captures webcam frames, performs face detection and recognition, analyzes head movement, and detects suspicious activities such as mobile phone usage, face absence, and multiple person presence. OpenCV, YOLOv8, and MediaPipe are integrated to ensure accurate and automated monitoring during online examinations. Whenever suspicious activity is identified, the system generates warning alerts, captures screenshots, stores activity logs, and updates the admin dashboard automatically. The methodology ensures secure, efficient, and intelligent online examination invigilation with automated evidence generation and reporting capabilities.

Initially, ride requests are submitted by passengers through frontend interfaces or APIs. The backend layer validates incoming requests, verifies passenger and driver information, processes ride details, and identifies nearby available drivers using GPS-based location tracking systems. This preprocessing stage improves data integrity and ensures standardized ride

management throughout the platform.

The intelligent ride allocation mechanism is integrated into the methodology to provide efficient driver-passenger matching. The backend processes ride parameters such as passenger location, driver availability, ride distance, estimated arrival

The AI Exam Invigilator System follows a real-time intelligent monitoring workflow designed to automate online examination invigilation using Artificial Intelligence and Computer Vision technologies.

Key Points of Workflow Methodology

- Secure student and admin authentication
- Real-time webcam initialization and monitoring
- AI-based face detection and recognition
- Continuous head movement and activity analysis
- Mobile phone detection using YOLOv8
- Automatic suspicious activity identification
- Real-time warning alert generation
- Screenshot evidence capture system
- CSV-based log storage and management
- Dashboard analytics and monitoring updates
- Automated PDF report generation
- Intelligent Computer Vision-based invigilation system

The methodology also incorporates secure system communication and access control mechanisms to ensure safe operation of the platform. Role-Based Access Control (RBAC) is implemented to restrict system functionalities based on user roles such as Passenger, Driver, Admin, and Super Admin. Additional security mechanisms including JWT authentication, Content Security Policy (CSP), secure API routing, and Cross-Site Request Forgery (CSRF) protection are integrated to prevent unauthorized access, malicious requests, and payment fraud [31].

Ride management and trip tracking functionalities are integrated into the workflow to improve operational coordination among users. Once a ride request is accepted, the platform continuously updates ride status, GPS location, estimated arrival time, and payment information in real time. The trip management module enables ride tracking through different ride states such as Requested, Accepted, In Progress, Completed, and Cancelled. This centralized workflow improves transportation efficiency and reduces manual coordination overhead [32].

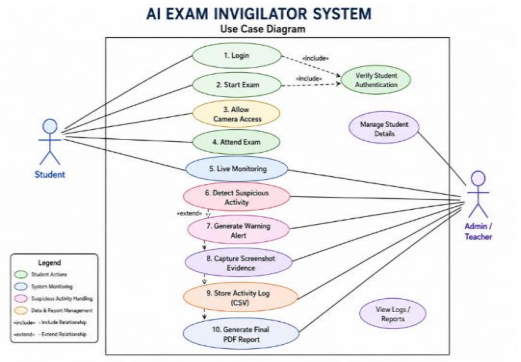
The proposed methodology also supports scalability and future extensibility. The modular system design enables integration of additional functionalities such as AI-based route optimization, cloud deployment, machine learning-based demand prediction, dynamic pricing systems, and real-time traffic analytics in future versions of the platform.

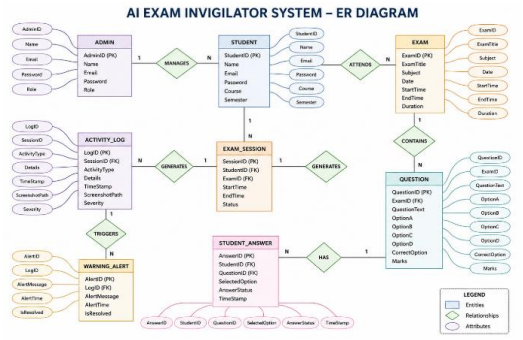
The structured methodology adopted in this platform improves ride allocation efficiency, reduces passenger waiting time, and enhances operational visibility compared to traditional transportation management systems.

- SYSTEM DESIGN AND IMPLEMENTATION

System defines the overall structure, architecture, modules, and data flow of the platform. The system is designed using a modular layered architecture that integrates frontend technologies, backend processing, Artificial Intelligence modules, and storage components into a unified monitoring platform. The design focuses on real-time examination monitoring, automated suspicious activity detection, secure authentication, and efficient evidence management.

- The system design includes various diagrams such as Use Case Diagram, ER Diagram, User Flow Diagram, System Architecture Diagram, and Data Flow Diagram to represent the interaction between users, system modules, AI components, and data storage processes. The modular design improves scalability, maintainability, security, and overall system performance while ensuring smooth integration between AI-based monitoring and web-based dashboard functionalities.
- The Use Case Diagram represents the interaction between users and the AI Exam Invigilator System. The main actors of the system are Student and Admin.
- **Student Functions**
- • Login to the system
  • Start examination monitoring
  • Face registration
  • Attend online examination
  • Webcam activation
- **Admin Functions**
- • Login to dashboard
  • Monitor suspicious activities
  • View activity logs
  • Access screenshots
  • Download PDF reportThe backend system is implemented using Flask REST APIs, which handle request processing, business logic execution, intelligent ride allocation, fare estimation, authentication, GPS synchronization, and database communication. Flask was selected due to its lightweight architecture, flexibility, and efficient API integration capabilities [35].





The Entity Relationship (ER) Diagram represents the database structure and relationships between different entities of the system.

Main Entities

Student Entity

Stores:

- student ID,
- name,
- login credentials,
- registration details.

Activity Logs Entity

Stores:

- suspicious activities,
- timestamps,
- activity type,
- evidence details.

Screenshot Entity

Stores:

- screenshot paths,
- timestamps,
- related activity information.

Report Entity

Stores:

- generated PDF reports,
- report details,
- monitoring summaries.

Admin Entity

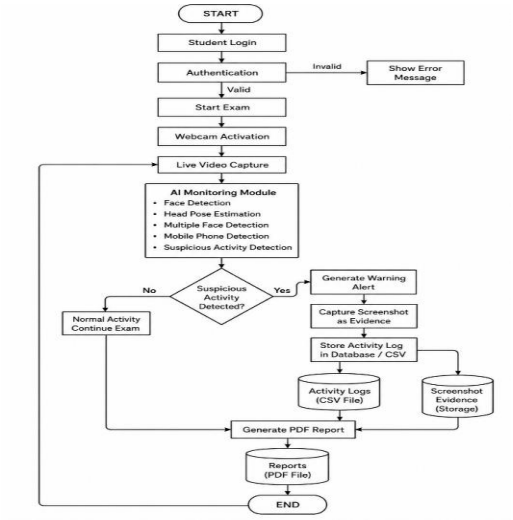
Stores:

- admin login credentials,
- authentication details.

The ER Diagram helps maintain proper data organization and relationship management within the System.

The Sequence Diagram represents the communication flow between frontend, backend, database, and system modules during various operations.

It explains how requests are processed step-by-step and how responses are generated for the user.



These visual analytics improve transportation visibility and support faster decision-making by administrators and operational managers.

1. The experimental User opens the website
2. User fills the inquiry form
3. Frontend validates form data
4. Data is sent to backend using POST request
5. PHP backend processes and sanitizes input
6. Data is stored in MySQL database
7. Success response is generated
8. Frontend displays confirmation message
9. Admin views inquiry through dashboard

The This process ensures smooth communication between user and administrator.

- **Chatbot Interaction Workflow**

Steps:

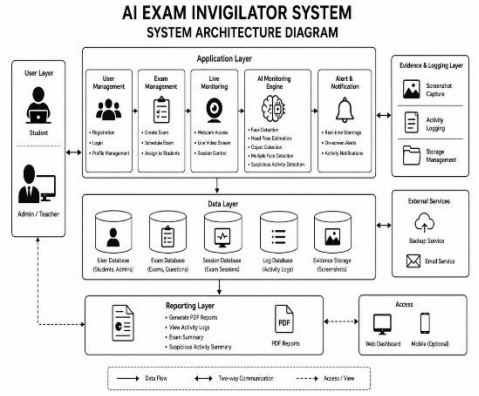
1. User interacts with chatbot
2. Chatbot collects user preferences
3. Frontend sends responses to backend
4. Backend processes user input
5. System matches stored logic and data
6. Appropriate response is generated
7. Output is displayed dynamically to user

- **frontend AI-Based College Predictor Workflow Steps:**

1. Student selects exam, branch, category, and rank
2. JavaScript dynamically filters available options
3. User submits prediction request
4. Frontend sends JSON request to predictor API
5. Backend validates and sanitizes data
6. Database query matches rank against cutoff ranges
7. Linear Probability Calculation is applied
8. Admission probability is generated
9. Results are categorized into:

- SAFE
- MODERATE
- DREAM

10. Dynamic result cards are displayed on frontend



This workflow transforms historical cutoff data into intelligent admission predictions. **Architecture Layers**

Frontend Layer

Technologies:

- HTML,
- CSS,
- Bootstrap,
- JavaScript.

Responsibilities:

- login interface,
- dashboard UI,
- monitoring display,
- alert visualization.

Backend Layer

Technologies:

- Flask,
- Python.

Responsibilities:

- route handling,
- AI integration,
- report generation,
- authentication.

AI Processing Layer

Technologies:

- OpenCV,
- YOLOv8,
- MediaPipe.

Responsibilities:

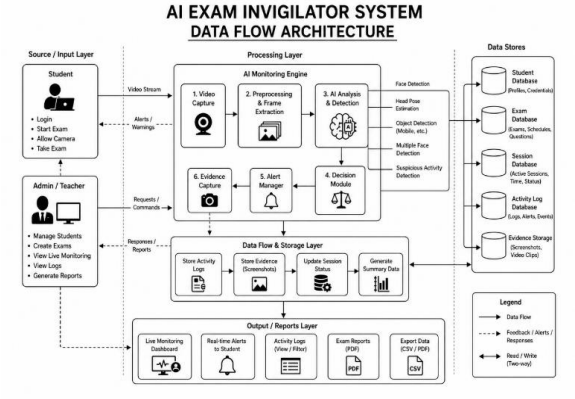
- face recognition,
- object detection,
- movement analysis,
- suspicious activity detection.

Storage Layer

Stores:

- CSV logs,
- screenshots,
- trained models,
- PDF reports.

The architecture ensures modularity, scalability, and efficient system performance.



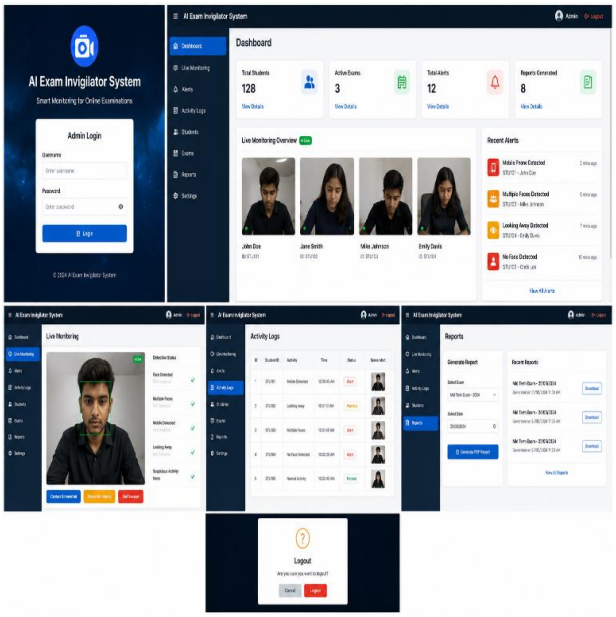
- **RESULTS AND DISCUSSION**

The AI Exam Invigilator System was tested under different examination conditions to evaluate its monitoring accuracy, real-time performance, security handling, and user experience. The system successfully monitored student activities using AI-based computer vision techniques and generated alerts for suspicious behavior during online examinations.

The testing process focused on face detection accuracy, mobile phone detection, warning generation, dashboard functionality, and report generation performance.

Functional testing was performed to verify whether each module of the system works correctly according to the expected functionality.

Module	Expected Result	Actual Result	Status
Student Login	User authentication	Successfully authenticated	Passed
Webcam Activation	Camera access enabled	Webcam activated correctly	Passed
Face Detection	Detect student face	Face detected accurately	Passed
Mobile Detection	Detect mobile phone	Mobile detected successfully	Passed
Warning Alert	Generate alert for suspicious activity	Alert generated correctly	Passed
Screenshot Capture	Capture evidence image	Screenshot stored successfully	Passed
CSV Log Storage	Save activity logs	Logs saved correctly	Passed
PDF Report Generation	Generate final report	PDF generated successfully	Passed



The face recognition module was tested under different lighting conditions and camera angles. The system successfully identified student faces and detected abnormal conditions such as missing faces or multiple persons.

Testing Conditions

- Normal lighting
- Low lighting
- Side face movement
- Multiple face appearance
- Face absence

Observed Results

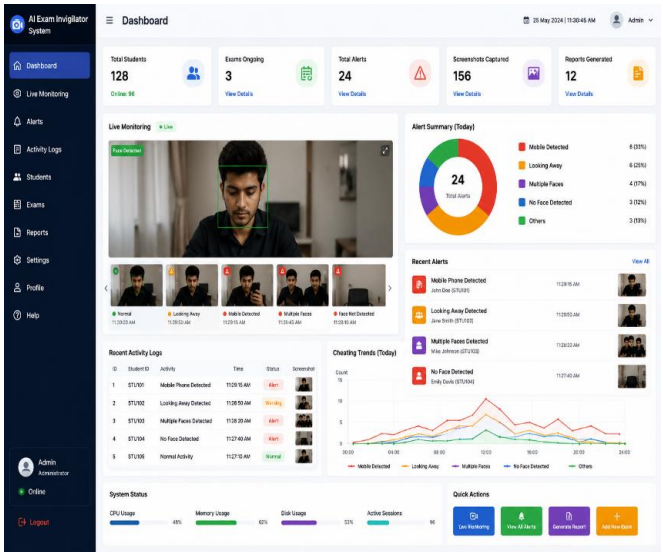
Test Scenario	Detection Result
Single student face	Successfully detected
Multiple faces	Suspicious activity detected
Face missing	Warning generated
Side movement	Head movement detected
Low light condition	Moderate accuracy

Analysis

The system achieved high detection accuracy in normal lighting conditions. However, detection performance slightly decreased in poor lighting environments.

The AI Exam Invigilator System was developed to improve online examination monitoring using Artificial Intelligence and Computer Vision technologies. During testing and implementation, several advantages, limitations, and practical applications of the system were observed.

The discussion section analyzes the system performance, monitoring capability, real-world usability, and comparison with traditional online examination monitoring systems.



REFERENCES

1.OpenCV, "Open Source Computer Vision Library," Available: OpenCV Official Website

2.Flask, "Flask Documentation," Available: Flask Official Documentation

3.YOLO, "YOLO Object Detection System," Available: Ultralytics YOLO

4.NumPy, "NumPy Documentation," Available: NumPy Official Website

5.Pandas, "Pandas Documentation," Available: Pandas Official Website

6.TensorFlow, "TensorFlow Documentation," Available: TensorFlow Official Website

7.Python, "Python Programming Language Documentation," Available: Python Official Website

8.Visual Studio Code, "Visual Studio Code Documentation," Available: VS Code Official Website

9.Computer Vision, Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2011.

10. Artificial Intelligence, Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education.

11. Machine Learning, Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education.

12. Face Recognition, Research Papers on Face Recognition and Object Detection Techniques, IEEE